Introdução

Hoje, os cursos da área de computação, Ciência da Computação, por exemplo, tem uma quantidade de meninos muito maior do que a quantidade de meninas. Motivada por essa estátistica, a Professora Maristela Terto de Holanda aplicou um formulário de 2011 a 2014 na Semana Nacional de Ciência e Tecnologia (SNCT) em Brasília buscando encontrar fatores que são impactantes quando um menina quer escolher um curso de computação. Os dados foram disponibilizados pelo professor Vinicius Ruela Pereira Borges.

Esse trabalho irá realizar as seguintes atividades:

1. **Limpeza dos Dados:** explicar como foi o processo de carregamento dos dados (o processo de extração foi previamente feito) e como os dados inconsistentes foram removidos.
2. **Classificação de Atributos:** explicar quais métodos foram usados para classificar a importância/seleção dos dados para a próxima etapa.
3. **Criação de um modelo estátistico**: usando o modelo *Support Vector Machine* [1], criar um modelo que mostre quais os atributos são mais importantes quando um menina vai decidir que quer ou não fazer o curso de Ciência da Computação.

Objetivo

Utilizar os dados para conseguir encontrar algum fator que tenha bastante influência quando meninas querem decidir qual curso superior querem fazer e se vão ou não fazer um curso na área de computação. Além disso, encontrar pontos que podem ser melhorados na formulário e assim facilitar estudos futuros.

Pacotes Utilizados

```
library(dplyr)
library(caret)
library(corrplot)
library(knitr)
```

Limpeza dos Dados

Inicialmente, o *data frame* possui as seguintes informações armazenadas.

```
df <- read.csv("data.csv", header = TRUE, na.strings=c(""),
                  stringsAsFactors=FALSE)
names(df)
```

```
##  [1] "Year"
##  [2] "Gender"
##  [3] "Educational.Stage"
##  [4] "Field.Of.Interest"
##  [5] "Would.Enroll.In.CS"
##  [6] "Q1"
##  [7] "Q2"
##  [8] "CS.Only.Teaches.To.Use.Software"
##  [9] "CS.Uses.Little.Math"
## [10] "Most.CS.Students.Are.Male"
## [11] "CS.Requires.Knowledge.In.Computers"
## [12] "Higher.Education.Required.To.Work.In.CS"
## [13] "Family.Approves.CS.Major"
## [14] "CS.Has.Low.Employability"
## [15] "CS.Work.Has.Long.Hours"
## [16] "CS.Fosters.Creativity"
## [17] "CS.Is.Prestigious"
## [18] "CS.Provides.Good.Wages"
## [19] "CS.Enables.Interdisciplinary.Experiences"
## [20] "Uses.Computer.At.Home"
## [21] "Uses.Computer.At.Relatives.House"
## [22] "Uses.Computer.At.Friends.House"
## [23] "Uses.Computer.At.School"
## [24] "Uses.Computer.At.Work"
## [25] "Uses.Computer.At.Lan.House"
## [26] "Uses.Computer.At.Library"
## [27] "Uses.Computer.At.Digital.Inclusion.Center"
## [28] "Has.Used.Text.Editor"
## [29] "Has.Used.Image.Editor"
## [30] "Has.Used.Spreadsheet"
## [31] "Has.Used.Database"
```

```
## [32] "Has.Used.Internet"
## [33] "Has.Used.Social.Network"
## [34] "Has.Used.Email"
## [35] "Has.Used.Games"
## [36] "Has.Used.For.Creating.Web.Pages"
## [37] "Has.Used.For.Development"
## [38] "Has.Used.Other.Softwares"
```

Para facilitar o processo de limpeza desses dados, as células do *data frame* que possuem respostas em branco ("") são lidas como NA e assim podem ser removidas.

```
df <- df[complete.cases(df),]
```

Observa-se que alunos que responderam os questinários colocaram seu "Gênero".

```
kable(df %>%
        group_by(Gender) %>%
        summarize(total = n()))
```

Gender	total
F	2957
M	9

Como é possível notar, há a presença de 9 meninos que serão removidos da análise.

```
df <- df[df$Gender == 'F',]
```

Análise de Características

É importante visualizar a coluna `Would.Enroll.In.CS` pois essa demonstra o interesse do estudante em curso um curso de Ciência da Computação.

Would.Enroll.In.CS	total
Maybe	1134
No	816
Yes	1007

As características importantes para a analise estão presentes na colunas 8 até a 38, as quais são perguntas do questionário que mostram atividades ou hábitos que podem possivelmente aumentar ou diminiur o interesse da estudante nos cursos de computação.

Para essa análise, vamos remover as colunas `Year`, `Gender`, `Educational.Stage`, `Field.Of.Interest`, `Q1`, `Q2` para simplificar a análise.

```
df <- df[, -(1:4), drop=FALSE ]
df <- df[, -(2:3), drop=FALSE ]
```

Preprocessamento

Para simplificar a analise, usa-se a matriz de correlação para identificar quais atributos estão muito relacionados entre si e assim podem ser removidos. Para essa análise, todos os valores devem ser números. A função `decide` analisa a resposta em:

1. “Yes” equivale a 2;
2. “Maybe” equivale a 1;
3. “No” equivale a 0

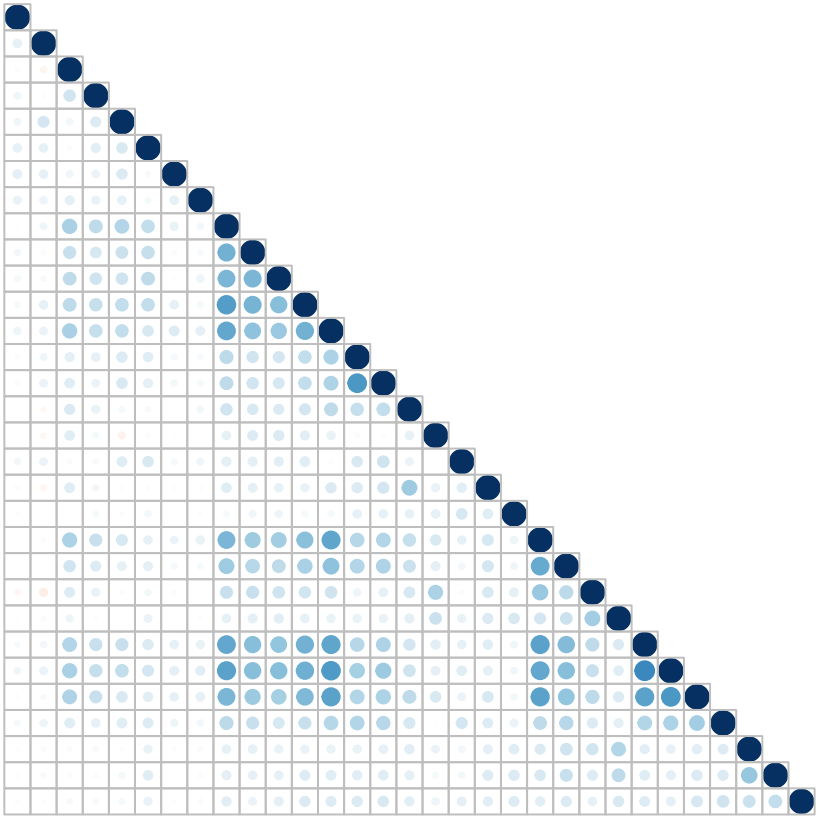
e troca o valor para número. A coluna `Would.Enroll.In.CS` será transformada em factor para análise de categorias.

```
decide <- function(x) {
  switch(x,
    "Yes"= {
      return(2)
    },
```

```
    "Maybe"={
      return(1)
    },
    "No"={
      return(0)
    },
    {
      return(0)
    }
  )
}

for(j in 1:32) {
  for(i in 1:3178) {
    df[i, j] <- decide(df[i, j])
  }
  df[,j] <- as.numeric(df[,j])
}

df$Would.Enroll.In.CS <- factor(df$Would.Enroll.In.CS,
                                levels=c(0,1,2),
                                labels=c("No", "Maybe", "Yes"))
```

Agora a matriz de correlação pode ser calculada:

```
corMatrix <- cor(df[,2:32])

corrplot(corMatrix, method = "circle", cl.pos = "n", tl.pos = "n",
                    type = "lower")
```



Observando o gráfico, existem poucos atributos correlacionados, mas ainda existem.

```
highlyCorrelated <- findCorrelation(corMatrix, cutoff=0.50)
print(names(df)[highlyCorrelated])
```

```
## [1] "Has.Used.Internet"
## [2] "Has.Used.Database"
## [3] "CS.Enables.Interdisciplinary.Experiences"
## [4] "Has.Used.Social.Network"
## [5] "Uses.Computer.At.Digital.Inclusion.Center"
## [6] "CS.Work.Has.Long.Hours"
## [7] "Uses.Computer.At.Relatives.House"
```

Remove-se as colunas com correlação maior que 0.5.

```
df <- df[,-highlyCorrelated]
```

Modelo de Treino

Para essa analise, o conjunto de dados será dividido em dois grupos: o grupo de treino e o grupo de testes.

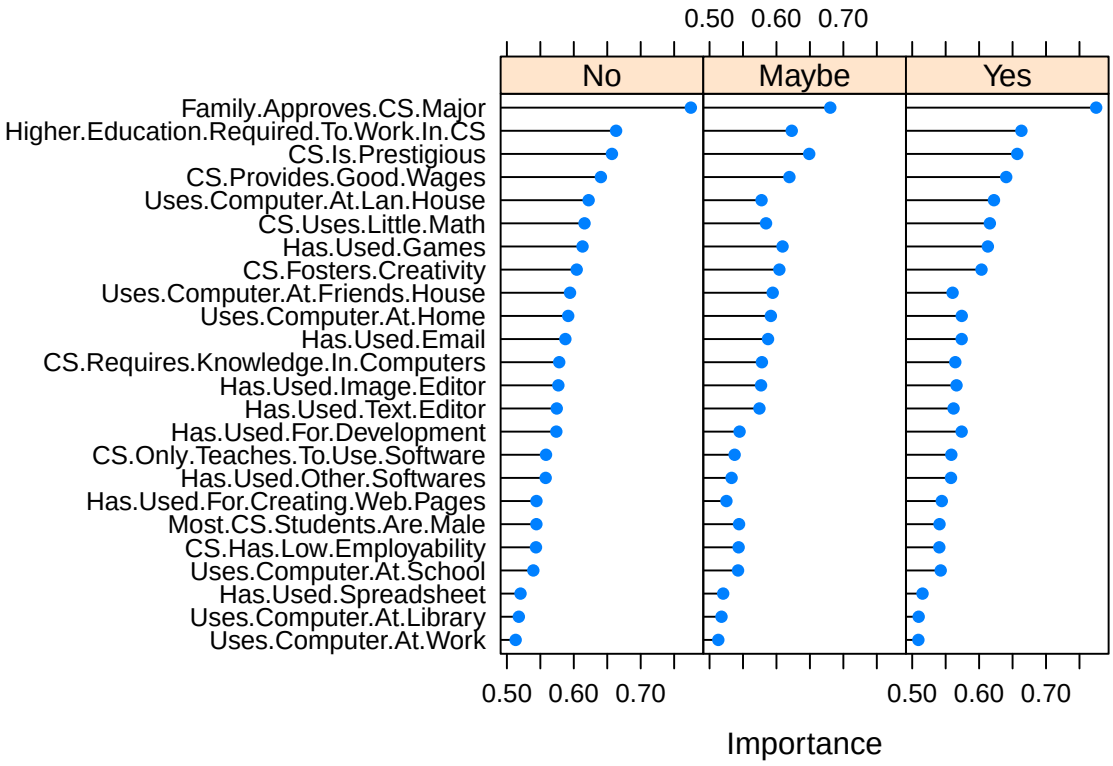
```
trainingIndexes <- createDataPartition(df$Would.Enroll.In.CS,
                                       p=0.85, list=FALSE)
trainingData <- df[trainingIndexes,]
testData <- df[-trainingIndexes,]
```

Usando o conjunto de dados de treino, usa-se a *Support Vector Machine* [1] com uma função polinomial como *kernel* para criar um previsão de como os dados se comportam.

```
trainingParameters <- trainControl(method="repeatedcv", number=10,
                                   repeats=2)

SVModel <- train(Would.Enroll.In.CS ~ .,
                 data = trainingData,
                 method = "svmPoly",
                 trControl= trainingParameters,
                 tuneGrid = data.frame(degree = 1,
                                       scale = 1,
                                       C = 1),
                 preProcess = c("pca","scale","center"),
                 na.action = na.omit
)
```

Com o modelo preparado, usa-se a função `varImp` para descobrir quais colunas tem um impacto maior em cada classificação, ou seja, se o candidato optou por "No", "Maybe" ou "Yes" quando respondeu `Would.Enroll.In.CS`.

```
importance <- varImp(SVModel, scale=FALSE)
plot(importance)
```



Agora, usando o conjunto de dados de teste, uma amostragem é criada tentando prever quantas respostas corretas podem ser alcançadas usando esse modelo.

```
predictions <- predict(SVModel, testData)
cm <- confusionMatrix(predictions, testData$Would.Enroll.In.CS)
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Maybe Yes
##      No    80    33  26
##      Maybe 62    89  41
##      Yes   13    48  84
##
## Overall Statistics
##
##                Accuracy : 0.5315
##                  95% CI : (0.4856, 0.5771)
```

```
##     No Information Rate : 0.3571
##     P-Value [Acc > NIR] : 6.443e-15
##
##                   Kappa : 0.2947
##  Mcnemar's Test P-Value : 0.003287
##
## Statistics by Class:
##
##                      Class: No Class: Maybe Class: Yes
## Sensitivity             0.5161       0.5235     0.5563
## Specificity             0.8162       0.6634     0.8123
## Pos Pred Value          0.5755       0.4635     0.5793
## Neg Pred Value          0.7774       0.7148     0.7976
## Prevalence              0.3256       0.3571     0.3172
## Detection Rate          0.1681       0.1870     0.1765
## Detection Prevalence    0.2920       0.4034     0.3046
## Balanced Accuracy       0.6662       0.5935     0.6843
```

Usando o Modelo *Support Vector Machine*, temos uma precisão de 56,51%.

Conclusão

Como a análise realizada nesse relatório, podem-se chegar a duas conclusões:

1. A partitipação da familía tem um impacto muito forte para a candidata escolher ou não um curso na área de computação.
2. As perguntas que tem uma importância menor 0.6 podem ter uma abordagem diferente para que em futuras pesquisas possam se aprofundar mais nessa questão e se aproximar mais de uma solução prática.

Referências

[1] https://pt.wikipedia.org/wiki/Máquina_de_vetores_de_suporte acessado em 05/12/2017.